

NASA TECHNICAL NOTE

NASA TN D-6533



NASA TN D-6533

C.1

0133370



TECH LIBRARY KAFB, NM

LOAN COPY: RETU
AFWL (DO/L
KIRTLAND AFB, N. M.

**PARAMETER IDENTIFICATION USING
A CREEPING-RANDOM-SEARCH ALGORITHM**

by Russell V. Parrish

Langley Research Center

Hampton, Va. 23365



0133370

NATL. TECHNICAL INFORMATION SERVICE

1. Report No. NASA TN D-6533		2. Government Accession No.		3. Document Catalog No.	
4. Title and Subtitle PARAMETER IDENTIFICATION USING A CREEPING-RANDOM-SEARCH ALGORITHM				5. Report Date December 1971	
				6. Performing Organization Code	
7. Author(s) Russell V. Parrish				8. Performing Organization Report No. L-7853	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365				10. Work Unit No. 136-62-01-03	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546				13. Type of Report and Period Covered Technical Note	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract A creeping-random-search algorithm is applied to different types of problems in the field of parameter identification. The studies reported are intended to demonstrate that a random-search algorithm can be applied successfully to these various problems, which often cannot be handled by conventional deterministic methods, and, also, to introduce methods that speed convergence to an extremal of the problem under investigation. Six two-parameter identification problems with analytic solutions are solved, and two application problems are discussed in some detail. Results of the study show that a modified version of the basic creeping-random-search algorithm chosen does speed convergence in comparison with the unmodified version. The results also show that the algorithm can successfully solve problems that contain limits on state or control variables, inequality constraints (both independent and dependent, and linear and nonlinear), or stochastic models.					
17. Key Words (Suggested by Author(s)) Creeping random search Parameter optimization Parameter estimation Optimization of stochastic models				18. Distribution Statement Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified		21. No. of Pages 35		22. Price* \$3.00

PARAMETER IDENTIFICATION USING A CREEPING-RANDOM-SEARCH ALGORITHM

By Russell V. Parrish
Langley Research Center

SUMMARY

A creeping-random-search algorithm is applied to different types of problems in the field of parameter identification. The studies reported are intended to demonstrate that a random-search algorithm can be applied successfully to these various problems, which often cannot be handled by conventional deterministic methods, and, also, to introduce methods that speed convergence to an extremal of the problem under investigation. Six two-parameter identification problems with analytic solutions are solved, and two application problems are discussed in some detail.

Results of the study show that a modified version of the basic creeping-random-search algorithm chosen does speed convergence in comparison with the unmodified version. The results also show that the algorithm can successfully solve problems that contain limits on state or control variables, inequality constraints (both independent and dependent, and linear and nonlinear), or stochastic models.

INTRODUCTION

The field of parameter identification, that is, the problem of selecting optimal values for parameters in a mathematical model representing some physical system, is rapidly expanding into all areas of design work, as well as into a variety of other engineering contexts. Examples of the problem arise in two-point boundary value problems, control-system design problems, dynamic-model synthesis, and network-design problems. The importance of parameter optimization in the engineering design process has been pointed out in reference 1.

In surveying the literature available in the field of parameter identification, two terms are encountered which seem to be used interchangeably: parameter optimization (refs. 1 to 4) and parameter estimation (refs. 5 and 6). A distinction is made between these two terms in the present paper, as follows:

Parameter optimization is concerned with determining the parameter values of a model that define an extremum of some cost function (also referred to as a performance

index or objective function) which is a function of model response. Problems in the area of mathematical programming and optimal control generally fall into this category. Parameter estimation is concerned with determining the parameter values of a model in such a way that the model response matches the response of an actual physical system or desired reference system. In parameter estimation, the cost function is usually a function of the difference between the model response and some previously obtained test data from the physical system. Problems in this category can generally be subdivided into two classes. The first is one in which the parameters have physical meaning, such as the determination of aerodynamic derivatives from flight-test data; and the second is one in which the parameters may have no meaning in the real world, such as curve fitting.

In any event, the procedure usually adopted to solve problems in either subclass is iterative in nature, involving perturbation of the parameters, evaluation of the cost function, a new perturbation of the parameters based on information gained from past perturbations, and so forth. Many deterministic algorithms exist in which these iterative methods are utilized; namely, steepest descent (ref. 7), conjugate gradient (ref. 8), Newton-Raphson (ref. 5), quasilinearization (ref. 6), Kalman filter (ref. 9), and simplex technique (ref. 10). These methods, as opposed to the random methods, generally converge well for favorable cost functions. However, as pointed out in references 4 and 11, if the cost function contains large higher-order derivatives with respect to the parameters, information gained from past perturbations will be of small value in determining future perturbations.

Perhaps more significant is the fact that deterministic methods often fail completely when physical limits are present on the control variables in the mathematical model (i.e., limits on commanded rates from an autopilot) and, also, when inequality constraints are imposed on the state variables and/or the parameters of the system.

Random methods, on the other hand, can be applied to problems containing any of the above restrictions. In reference 4, sequential random perturbation, or creeping random search, is suggested as a likely candidate for a successful algorithm that can be applied to many different classes of problems. However, the creeping-random-search algorithm is a brute-force method, in which many solutions of the mathematical model are required; as such, this algorithm is limited in its application by the number of parameters to be selected and the amount of time necessary to solve the mathematical model.

This report will introduce a modified version of the creeping-random-search algorithm that reduces the possible number of cost-function evaluations and still has the advantage of being applicable to many different types of problems (in particular, to those problems containing limits or inequality constraints). In addition, some of the various strategies available within the algorithm for speeding convergence to an extremum are evaluated and shown to be highly problem dependent in regard to successful application.

Another method for speeding convergence, which utilizes the interactive features provided by a computer facility, is introduced and demonstrated. The algorithm was developed for use on the NASA Langley Research Center's CDC 6600 real-time digital simulation subsystem described in reference 12, in both an all-digital mode of operation and a hybrid mode of operation, in conjunction with a GPS 10 000 iterative analog computer.

SYMBOLS

A_z	stored flight data (normal acceleration)
a_z	normal acceleration of attacking aircraft
a_i, b_i, b_j, c, k	arbitrary constants
F_j	arbitrary functional
F_p	penalty function
g	gravitational constant
$J(\alpha_1, \alpha_2)$	scalar cost function dependent upon parameter values
K_4	constant
n	number of parameters (see table I)
P	probability (see table I)
p_A	roll rate of attacking aircraft
R	relative displacement between two aircraft
r	number of tries (see table I)
s	Laplace operator
t	time
u	out-of-plane displacement

V_A	velocity of attacking aircraft
V_T	velocity of target aircraft
v	horizontal displacement
v_{in}	horizontal displacement at entrance of magnetic field
v_{out}	horizontal displacement at exit of magnetic field
x_A	range of attacking aircraft
\dot{x}_A	range rate of attacking aircraft
x_T	range of target aircraft
\dot{x}_T	range rate of target aircraft
y_A	lateral displacement of attacking aircraft
\dot{y}_A	lateral velocity of attacking aircraft
y_T	lateral displacement of target aircraft
\dot{y}_T	lateral velocity of target aircraft
z	vertical displacement
$\frac{du(v_{in})}{dv}$	out-of-plane slope at v_{in}
$\frac{dz(v_{out})}{dv}$	vertical slope at v_{out}
α_i	unknown parameter
δ_a	aileron-deflection command
δ'_a	limited aileron-deflection command

λ	line-of-sight angle from horizontal
l, m, \dot{l}, \dot{m}	dummy variables
ϕ_A	roll angle of attacking aircraft
$\dot{\phi}_A$	roll rate of attacking aircraft
$\phi_{c,A}$	roll-angle command of attacking aircraft
ψ_A	yaw angle of attacking aircraft
$\dot{\psi}_A$	yaw rate of attacking aircraft
$\psi_{c,A}$	yaw-angle command of attacking aircraft
$\dot{\psi}_{c,A}$	yaw-rate command of attacking aircraft
$\dot{\psi}_{c,A}^*$	limited yaw-rate command of attacking aircraft
ψ_T	yaw angle of target aircraft

ANALYSIS OF CREEPING-RANDOM-SEARCH ALGORITHM

The basic algorithm is illustrated with the two-parameter contour plot shown in figure 1. This basic algorithm, devoid of any strategies, operates in the following manner: In this simple problem, where the cost function to be minimized is $J(\alpha_1, \alpha_2) = \alpha_1^2 + \alpha_2^2$, constant values of the cost function are represented as concentric circles in the two-parameter plane. From an arbitrary starting point, a random step is taken and the cost function is evaluated. If the step has improved the cost function (a success), the present parameter values are used as a new starting point and a new step is taken. If the cost function was not improved (a failure), the original parameter values are retained and a new step is taken. Thus, a walk over the contour is generated.

In conjunction with the basic algorithm, four strategies (to be explained in more detail later) were used:

- (1) Last successful direction first; after each success, the next step is taken in the same direction
- (2) Variable step size based on runs; the step size is increased after several successes in a row or decreased after several successive failures

- (3) Correlation on past history; future steps favor the general direction in which past improvements were obtained
- (4) Correlation on last success; future steps favor the general direction in which the last improvement was obtained

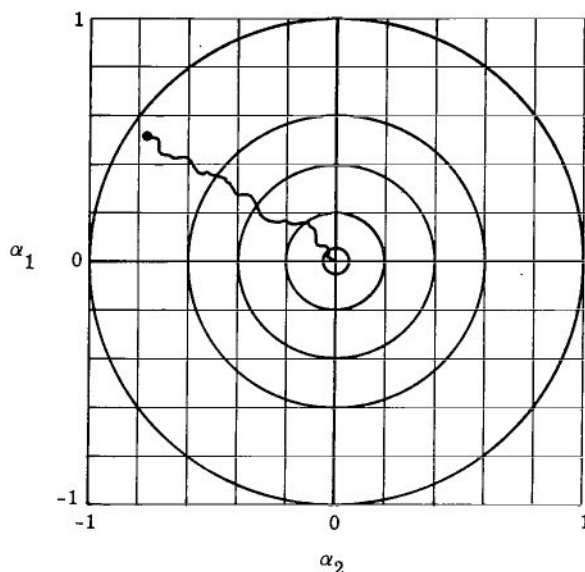


Figure 1.- Two-parameter contour plot of basic algorithm.

In explaining the basic algorithm in more detail, attention is directed to figure 2. As illustrated in figure 2, there are nine possible directions that can be taken from a given starting point in the two-parameter plane. These nine directions are obtained by plus, minus, and zero perturbations of each parameter. (Note that one direction is no movement at all.) The eight directions that involve movement are stored in an array

Sign	Designation
+	2
0	1
-	0

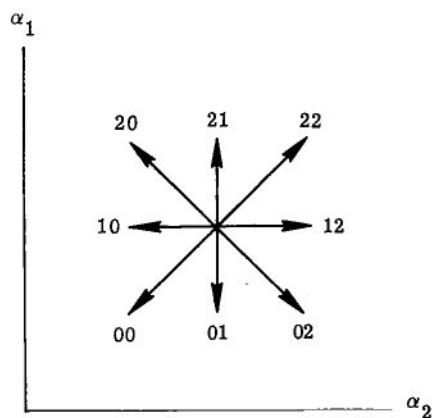


Figure 2.- Details of basic algorithm.

which the algorithm samples randomly (ref. 13) without replacement until a success is obtained or all directions are exhausted, at which point the array is reinitialized. The storage designations for the directions are shown in figure 2. Note that the designations for opposite directions sum to 22. This fact was used to eliminate one direction from consideration whenever a success occurred; namely, the direction opposite to that which obtained the success (which is automatically a failure). Naturally, this direction is returned to consideration when the table is reinitialized.

If failures should occur in all possible directions from a given point, the algorithm has either converged to an extremum, overstepped an extremum, or overstepped a pass (a pass is analogous to a ridge in a maximization problem). Figure 3(a) illustrates the case of overstepping an extremum; a reduction in step size would be necessary in order to continue convergence. Figure 3(b) demonstrates the case of overstepping a pass; and again a reduction in step size is necessary in order to continue convergence.

Thus the event of failures in all possible directions from a given point requires a reduction in step size. This fact led directly to the comparison of two concepts of the basic algorithm; namely, a redundant algorithm (as proposed in ref. 4 for use on a highly specialized hybrid computer) and a nonredundant algorithm (the modified-creeping-search algorithm). In the redundant algorithm, no memory of past directions used from a given point is available; and, therefore, the same direction may be tried several times. The nonredundant algorithm has memory and never tries the same direction twice with the same step size. Table I shows a comparison of the number of tries necessary before a reduction in step size can be made between these two basic concepts (ref. 14).

Again, a reduction in step size is necessary when overstepping has occurred or, equivalently, when failures are encountered in all possible directions. With the redundant

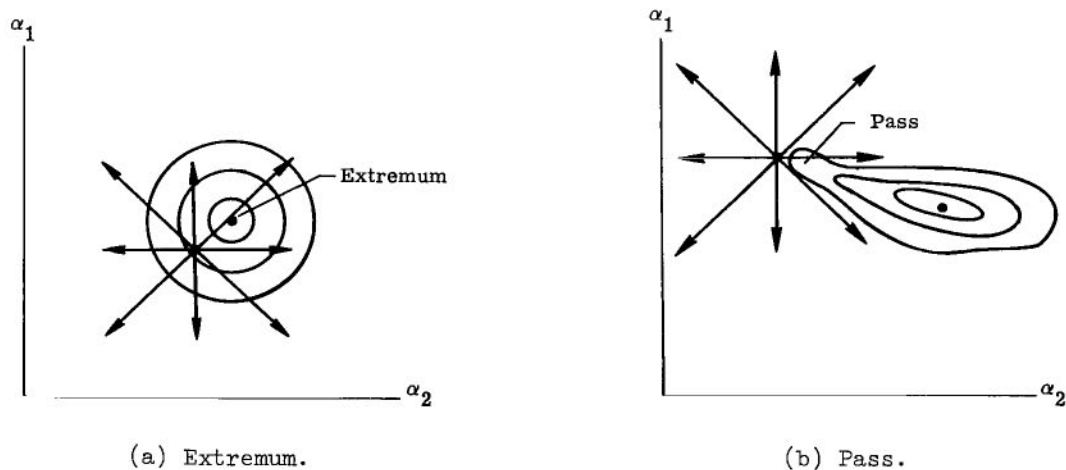


Figure 3.- Overstepping.

algorithm, there can never be the certainty that all possible directions have been tried. Therefore, a 90-percent confidence level on exhausting all of the directions has been set as the basis for comparison.

Examination of table I reveals that the number of tries necessary for both algorithms increases rapidly with an increase in the number of parameters. More importantly, the ratio of redundant to nonredundant tries also increases. Use of a variable step-size strategy (i.e., cut the step size after a number of failures in a row), which is employed in reference 4, may reduce the number of tries necessary when an overstep is encountered. However, this strategy can slow convergence significantly when a success could have been obtained in only a few of the many possible directions. Also, the successful use of this strategy is highly problem dependent, as will be demonstrated in the next section.

PERFORMANCE ANALYSIS OF NONREDUNDANT ALGORITHM

Two distinct groups of example problems (all two-parameter optimization problems with analytic solutions) were solved by using the nonredundant algorithm, with various combinations of the available strategies, in order to evaluate the performance of the method. The first group consisted of two static problems (algebraic equations) and one dynamic problem (differential equations). The problems of this first group were used for strategy evaluation. The second group consisted of three nonlinear programming problems (inequality parameter constraints and a nonlinear cost function), two of which are static problems and the third is a dynamic problem. The problems of this second group were used to demonstrate the application of the algorithm to constrained problems.

Strategy Evaluation

The problems of the first group were used to evaluate the different strategy combinations available, as well as any strategy parameters (i.e., the number of successes in a row to be required before increasing the step size under the variable step-size strategy). Conventional deterministic methods handle these problems readily.

Before discussing the problems and results, however, some description of the strategies and their interaction should be presented. Table II lists the combinations of strategies used. The first strategy, pure random, is simply the basic algorithm. The second strategy, last successful direction first, is the basic algorithm with the restriction that the first direction tried immediately following a success is in the same direction previously used. If use of this direction results in a failure, the remaining directions are sampled in the standard manner. The third strategy, variable step size based on runs, has no influence on the directions selected but, rather, controls the step size of

the parameter perturbations and the reinitialization of the array of directions. Whenever a step-size change occurs, the array is reinitialized. The fourth and fifth strategies, correlation on past history and on last success, respectively, operate essentially in the same manner as strategy 2, except that several directions (the correlated directions) are tried before returning to standard sampling in the event of failures. The correlated directions include all directions that are less than $\pm 90^\circ$ away from the general direction of past improvements for strategy 4 and the last successful direction for strategy 5.

Strategy combinations obey the following hierarchy now presented: 2, 4 or 5, 3. For example, strategy 11, consisting of the combined strategies 2, 3, and 4, would operate in the following manner after a success:

- (1) The first direction tried would be in the same direction as that in which the success was obtained.
- (2) If the first direction tried resulted in a failure, the next direction to be tried would be the direction in which past improvements were obtained (the trend direction).
- (3) If the second direction resulted in a failure, the remaining correlated directions would be tried.
- (4) If a success had still not been obtained, random sampling of the remainder of the directions would begin.
- (5) If at any point in the above procedure, the number of failures in a row exceeds the run parameter of strategy 3, the step size is cut, the array is reinitialized, and strategies 2 and 4 are discontinued until a success is obtained.

Table III displays the rank each strategy achieved, the rank being based on the average number of iterations required for acceptable convergence for each of the following problems (the average presented is based on 10 trails from the same point for each strategy, although averages were obtained from more than one starting point for dynamic problem I):

Static problem I (fig. 4):

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = \alpha_1^2 + \alpha_2^2$$

which is subject to

$$-1 \leq \alpha_i \leq 1 \quad (i = 1, 2)$$

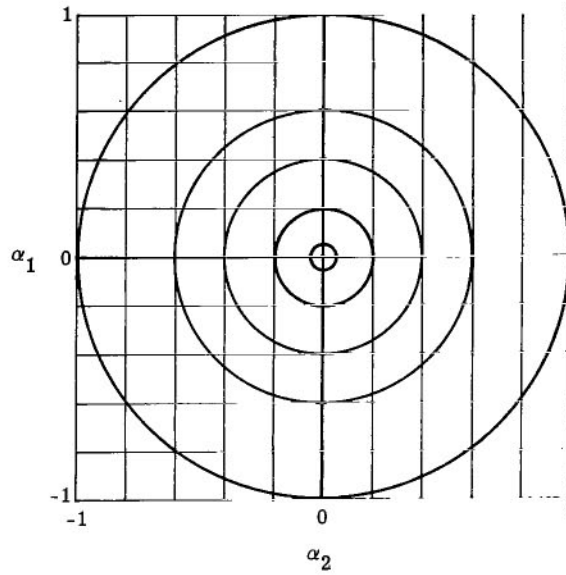


Figure 4.- Static problem I.

Static problem II (fig. 5):

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = \frac{(\alpha_1 + 0.1)^2}{0.25} + \frac{(\alpha_2 - 0.2)^2}{10}$$

which is subject to

$$-1 \leq \alpha_i \leq 1 \quad (i = 1, 2)$$

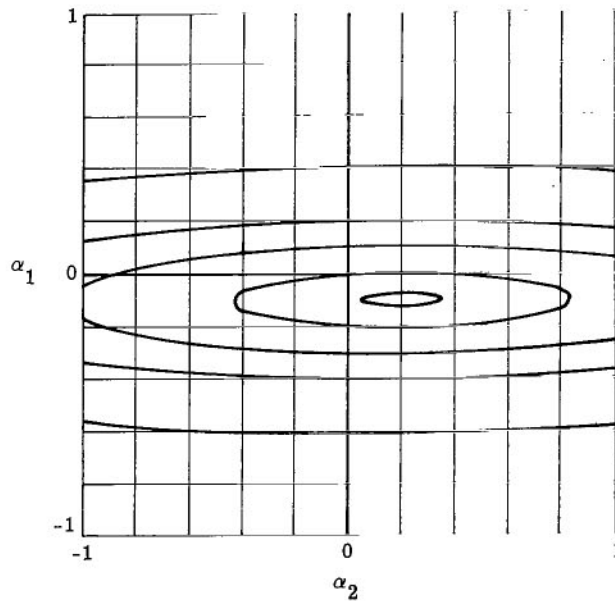


Figure 5.- Static problem II.

Dynamic problem I (fig. 6):

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = \int_0^2 \left[(0.8 \cos 3t - \alpha_2 \cos 10\alpha_1 t)^2 + \frac{1}{2}(0.3 - \alpha_1)^2 \right] dt$$

which is subject to

$$-1 \leq \alpha_i \leq 1 \quad (i = 1, 2)$$

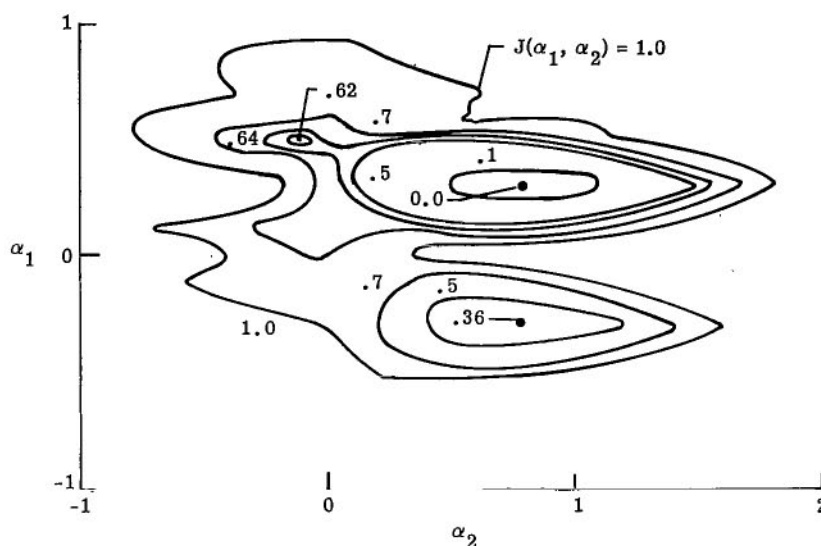


Figure 6.- Dynamic problem I.

It should be mentioned that dynamic problem I was solved in a hybrid environment. The only difficulty arising in the hybrid implementation was one of resolution. Either rescaling the analog portion of the program in vastly differing regions of cost-function values is necessary, or some minimum resolution on the parameter values must be accepted and the minimum step size allowed for parameter perturbations must be limited.

A review of table III demonstrates that strategy performance is problem dependent. However, strategy 2 (last successful direction first) and strategy 3 (variable step size based on runs) appear to be effective when used in combination with themselves and/or the other strategies.

Table IV lists the results of the run-parameter studies for the two static problems. Again, the results (number of iterations to convergence based on averages from 10 trials) appear to be highly problem dependent, even in the two-parameter domain. This situation would logically be intensified in a multiparameter domain.

Due to the high degree of problem dependence exhibited by both the run parameters and strategy performance, another strategy that utilizes analyst-algorithm interaction to speed convergence will be introduced later in the present study.

Constrained Problems

The problems of the second group were attempted in order to demonstrate the application of the algorithm to problems subject to inequality constraints. Before discussing the problems, however, it is necessary to clarify the use of the term "inequality constraints." All three problems of the first group were subject to inequality constraints of the type shown in the equation

$$-a_i \leq \alpha_i \leq b_i \quad (i = 1, 2, \dots, n) \quad (1)$$

Random-search methods handle this type of constraint without difficulty (steps across boundaries are simply not allowed), even in the region of a boundary where deterministic methods using a penalty-function approach have difficulty. For the purposes of this paper, a constraint in the form of equation (1) is referred to as an independent inequality parameter constraint; whereas a constraint in the form of the equation

$$F_j(\alpha_1, \alpha_2, \dots, \alpha_n) \leq b_j \quad (j = 1, 2, \dots, m) \quad (2)$$

is referred to as a dependent inequality parameter constraint. The dependent constraint may be termed linear or nonlinear, depending on the functional form of F_j .

The problems of the second group will now be discussed. Figure 7 displays the contour plot and the feasible region of solution for nonlinear static problem I as follows:

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = (\alpha_1 - 0.5)^2 + (\alpha_2 - 0.2)^2$$

which is subject to

$$-1 \leq \alpha_i \leq 1 \quad (i = 1, 2) \text{ (independent constraints)}$$

$$\alpha_1 + 2\alpha_2 \leq 0.6 \quad \text{(linear dependent constraint)}$$

The creeping-search algorithm was implemented by using a penalty function for violation of the dependent inequality constraint as follows:

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = (\alpha_1 - 0.5)^2 + (\alpha_2 - 0.2)^2 + F_p$$

where

$$F_p = \begin{cases} 5, & \alpha_1 + 2\alpha_2 > 0.6 \\ 0, & \alpha_1 + 2\alpha_2 \leq 0.6 \end{cases}$$

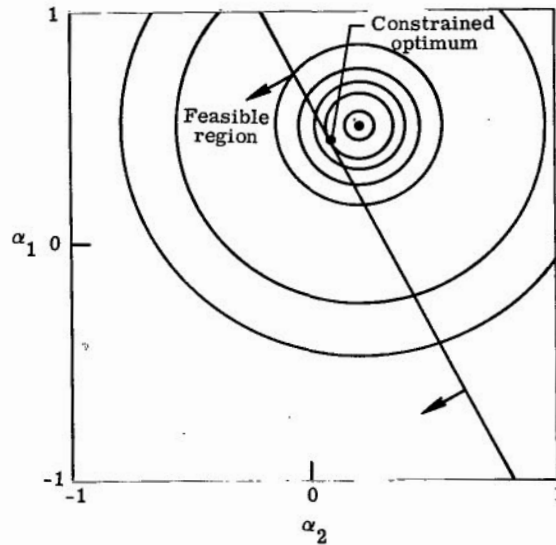


Figure 7.- Nonlinear static problem I.

The algorithm converged, not to an extremal of the constrained problem, but rather to some point on the boundary formed by the dependent constraint; the reason for this behavior can be deduced from figure 8. Obviously, no matter what step size is used in the indicated convergence region of figure 8(a), steps in the eight possible directions will result in either constraint violations or larger cost-function values. However, if the boundary encounter had occurred outside of the region as shown in figure 8(b), convergence would be possible as long as the walk approached the optimal from below.

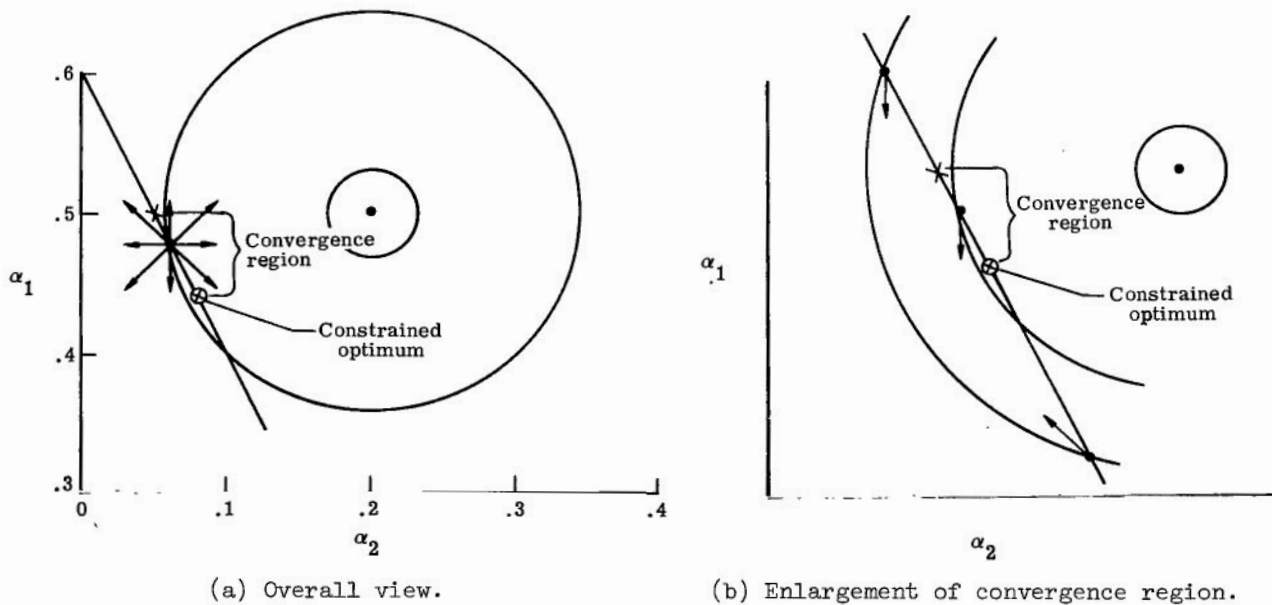


Figure 8.- Boundary convergence.

A simple adaptation of the algorithm that solves the above difficulty is to allow the algorithm two additional directions once a boundary is encountered; that is, one step in one direction along the dependent constraint, and the second step in the opposite direction along the constraint. This adaptation proved successful in solving this problem, as well as the following problems:

Nonlinear static problem II (fig. 9):

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = \alpha_1^2 - 2\alpha_1 - 2\alpha_2$$

which is subject to

$$\alpha_i \geq 0 \quad (i = 1, 2) \text{ (independent constraints)}$$

$$\left. \begin{array}{l} 2\alpha_1 + 6\alpha_2 \leq 6 \\ 2\alpha_1 + 2\alpha_2 \leq 4 \end{array} \right\} \text{ (linear dependent constraints)}$$

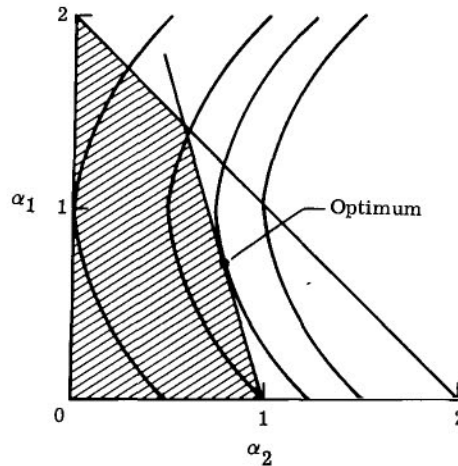


Figure 9.- Nonlinear static problem II.

Nonlinear dynamic problem I (fig. 10):

$$\min_{\alpha_1, \alpha_2} J(\alpha_1, \alpha_2) = \int_0^2 \left[(0.8 \cos 3t - \alpha_2 \cos 10\alpha_1 t)^2 + \frac{1}{2} (0.3 - \alpha_1)^2 \right] dt$$

which is subject to

$$\left. \begin{array}{l} -0.5 \leq \alpha_2 \leq 1 \\ 0 \leq \alpha_1 \leq 0.4 \end{array} \right\} \text{ (independent constraints)}$$

$$\alpha_2^2 + 2\alpha_1 \leq 1 \text{ (nonlinear dependent constraint)}$$

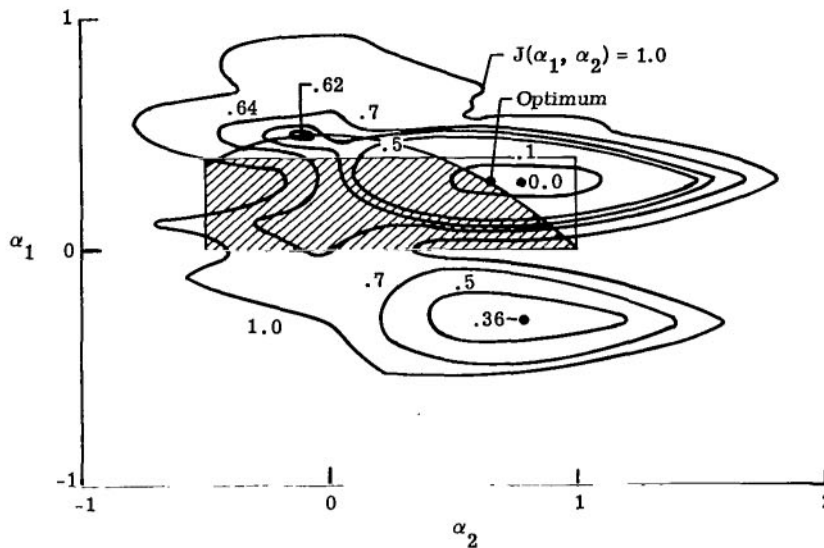


Figure 10.- Nonlinear dynamic problem I.

Again, the dynamic problem was solved in a hybrid environment, with an additional resolution problem added by the necessity of being able to detect convergence to the boundary in order to allow the addition of the two directions along the boundary. Convergence to the boundary was possible only to within the neighborhood of the boundary dictated by the minimum step size allowed. Therefore, detection of a boundary encounter was considered accomplished when convergence to within the neighborhood took place.

It should be noted that solutions to nonlinear static problems, with or without inequality constraints, are well documented in the literature (ref. 15).

Applications

Two investigations which have been made using the algorithm are a five-parameter human transfer function piloting an aircraft in a simple aircraft-pursuit problem (parameter estimation) and a five-parameter beam-transport-design problem (parameter optimization). Before discussing these problems and some results, however, a description of the strategy utilizing analyst-algorithm interaction should be presented.

The intended emphasis of the description is on the generation of the array of directions and on the cathode-ray-tube (CRT) display capabilities. These two features are used jointly, hopefully to speed convergence, in the following manner: The analyst may, during algorithm operation, display on the CRT a history of the cost-function behavior versus number of iterations (both successes and failures), as well as histories of the value of each parameter versus number of successes (also available would be displays related to the model). Based on this information, the analyst may be able to determine the sensitivity of certain parameters in the current region of operation of the cost domain.

If a parameter appears sensitive, the gain, or perturbation size, of that parameter may be increased. On the other hand, if a parameter appears insensitive (varies randomly about some mean) or if the parameter requires a certain value in order to obtain a success, it may be made constant. In this situation, the number of possible directions is greatly reduced, as evidenced by table I, and the array of directions is regenerated for the reduced number of parameters involved. Use of this method to speed convergence is illustrated in figures 11 and 12.

Figure 11 shows typical variations of parameters as a function of the number of successes from a seven-parameter study. The first parameter α_1 appeared to be quite sensitive in this region of the parameter space. Therefore, the analyst decided to increase the gain (the perturbation size) for this parameter. When the behavior of the parameter stabilized (became insensitive), it was made a constant. Little can be said of the sensitivity of α_2 , although initially the analyst decided to increase its gain. The third parameter α_3 appeared to be insensitive in this region (note the difference in

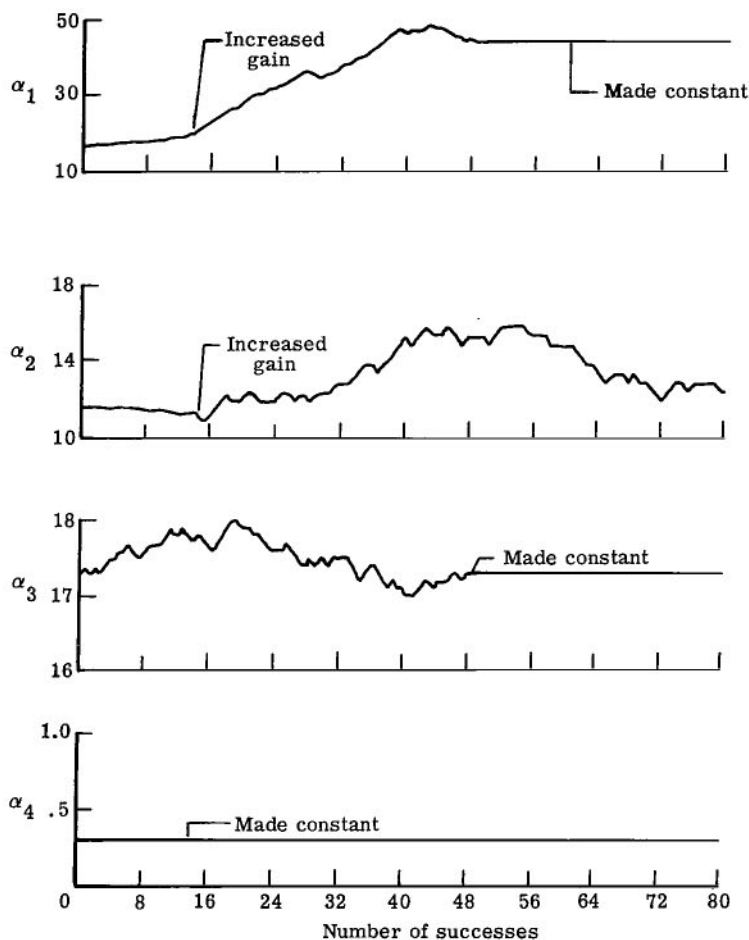


Figure 11.- Typical parameter histories.

scales), whereas α_4 seemed to require a certain value to obtain successes with the present step size. Thus the analyst chooses to fix the values of some parameters and to increase or decrease gains on other parameters in order to speed convergence in the current region of the cost-function domain.

The results of the analyst's decisions just discussed are displayed in figure 12 in terms of the cost function versus the total number of tries (both successes and failures). The analyst employed his initial decisions on the fifteenth success after 71 tries. As can be seen, successes occurred more often and with greater reductions in the cost function. This method was used in the following two application problems.

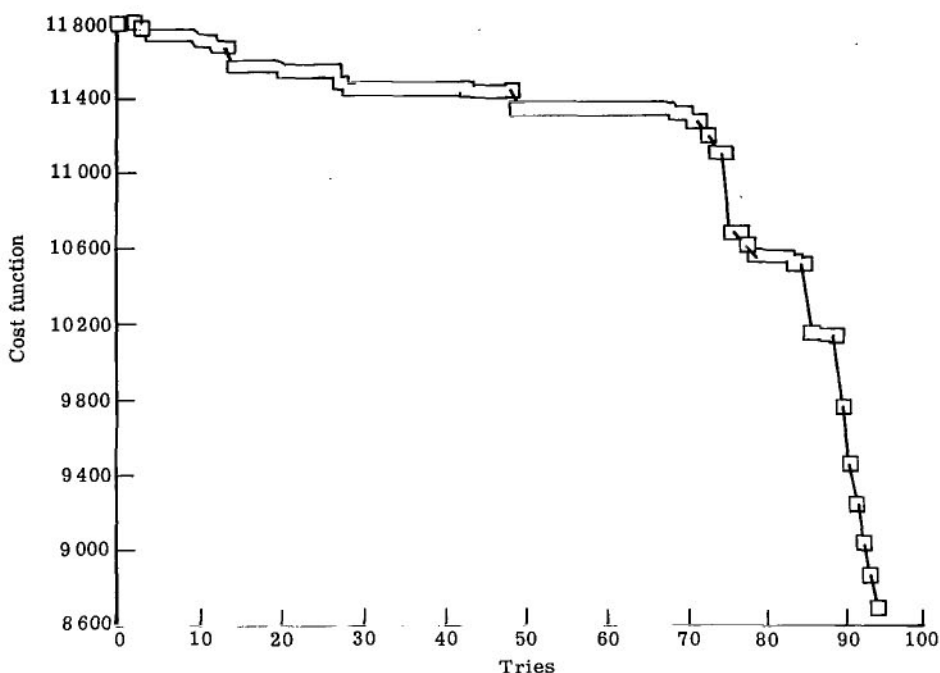


Figure 12.- Typical cost-function history.

Human-transfer-function problem.- As pointed out in reference 16, mathematical models for human pilots are beginning to play a large role in aerodynamic investigations. The present example is concerned with estimating the five parameters contained in a given transfer function used to pilot a simple three-degree-of-freedom aircraft model in a pursuit environment with a three-degree-of-freedom target (equations for this problem are presented in appendix A). The physical situation is represented in figure 13, and a block diagram of the system is illustrated in figure 14. Motivation for using the modified creeping-random-search algorithm for this problem was the presence of limits on transfer-function variables.

The parameters were to be estimated using a least-squares cost function consisting of differences between model response in the translational degrees of freedom and in normal acceleration a_z and actual flight data available for these variables. However, in order to eliminate from consideration such things as measurement noise (on flight data)

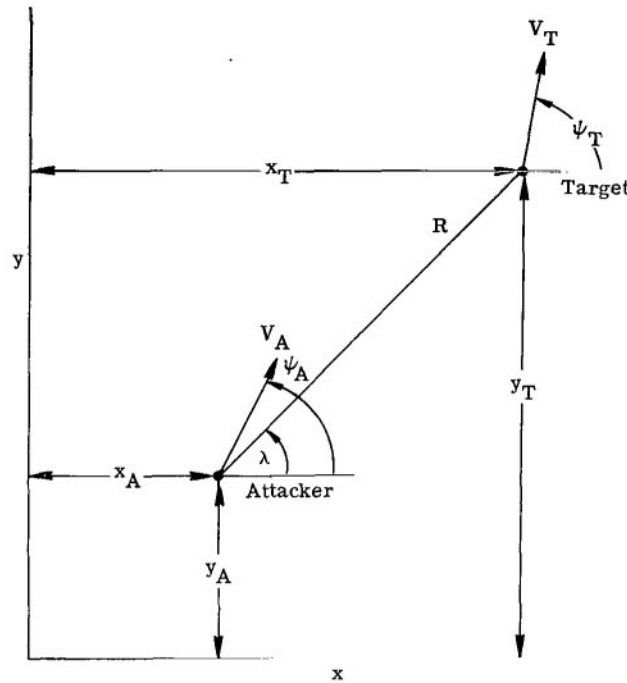


Figure 13.- Geometry of human transfer function problem.

and process noise (modeling error), as discussed in references 17 to 19, pseudoflight data – or flight data obtained from the model itself for selected parameter values – were used for the purposes of this paper. Only the normal-acceleration data were used in the estimation process.

Table V contains results obtained from the creeping-random-search algorithm, a steepest-descent algorithm, and a Davidon-conjugate-gradient algorithm for the same starting point. (The latter two algorithms used were part of the program of reference 7, and no attempt was made to alter the algorithms to handle the limits present in the mathematical model.) Tabulated are the true parameter values, and for each algorithm, the starting values, the final values, the number of cost-function evaluations, and the total computer time (central-processing-unit usage). The results indicate that the presence of limits in the mathematical model degrade the gradient information required by both the steepest-descent algorithm and the Davidon-conjugate-gradient algorithm to such a degree that convergence to the proper parameter values are extremely time consuming, if indeed possible.

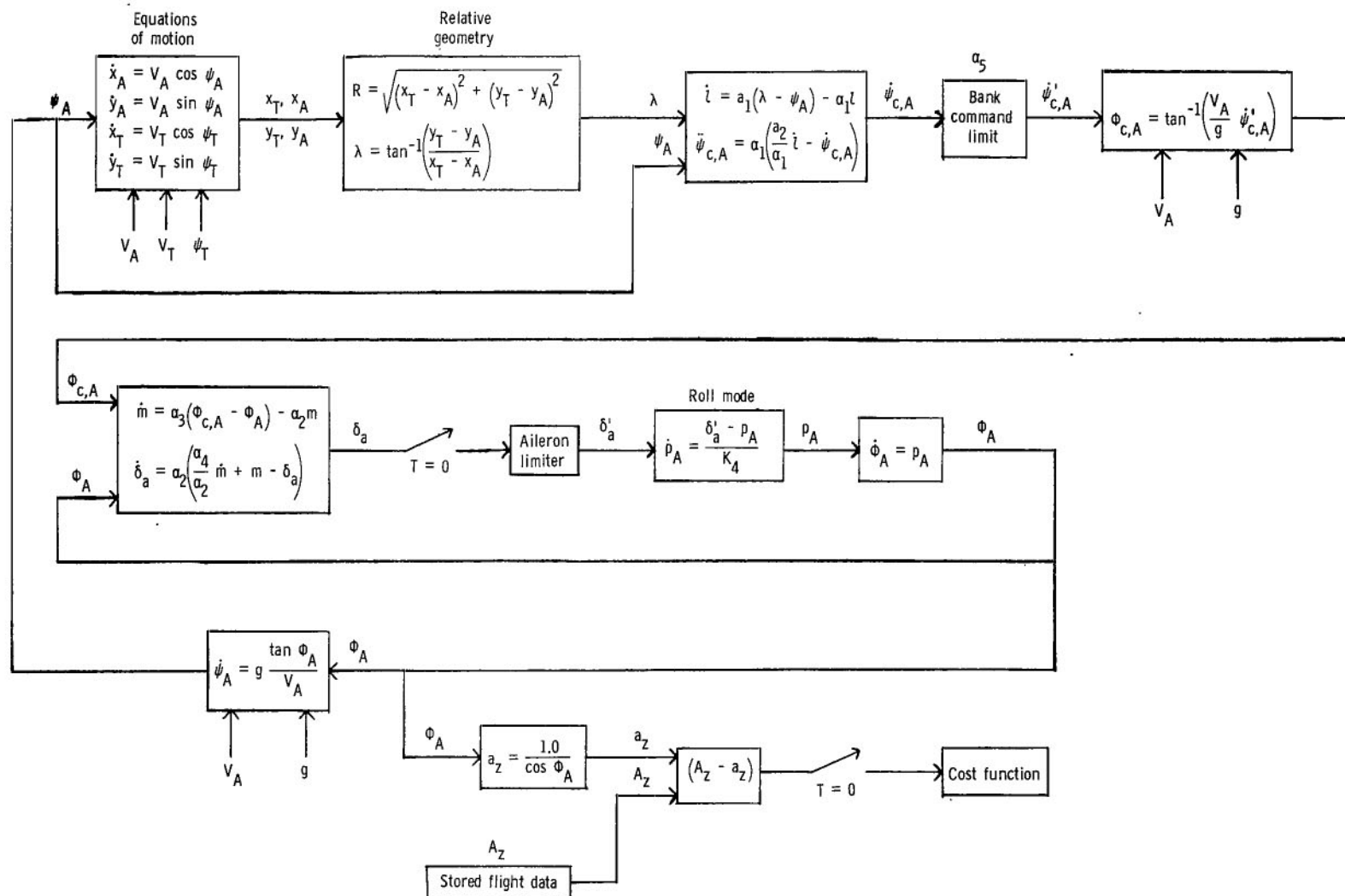


Figure 14.- Block diagram of human-transfer-function problem.

Beam-transport-design problem.- A beam-transport-design problem is concerned with transporting a high-energy beam of charged particles from one given area to another by utilizing magnetic fields. As reference 20 emphasizes, detailed computer-optimization solutions are highly desirable because of the large amount of effort and time that must be devoted to moving the unwieldy magnets. The fact that the problem inherently contains constraints on several state variables, as well as a stochastic model for representing individual members of the beam, often renders conventional deterministic optimization methods useless.

From an a priori probability distribution, used to model the position of individual members of the beam prior to entering the magnetic field, a set of random initial conditions for the state variables is selected. The equations of motion are then solved to evaluate the deviations from the desired end conditions and, also, to evaluate any violations of the state-variable constraints. This process is then repeated and the deviation is accumulated to form the mean cost function over the desired sample size. A constraint-violation percentage over the sample is also calculated and may be included in a weighted cost function if so desired. The parameters of the system, focusing strengths and separation distances of the magnets, are then perturbed by the creeping-search algorithm; and the entire sequence is iterated until convergence is achieved. A pooled t test, as described in reference 21, is used to determine successes or failures of parameter perturbations, thus fixing a confidence level on differences in cost-function means.

The five-parameter problem considered here used algebraic solutions, rather than numerical-integration solutions, of the linear differential equations of motion (nonlinear equations of motion are often required for other applications). The algebraic equations apply to the physical situation illustrated in figure 15, and all equations are presented in appendix B. A more complete treatment of the problem is found in reference 22.

Again extensive use of the CRT display capabilities is made, not only for plots of the cost function versus number of iterations and time histories of parameter values for successful tries, but also for model displays such as output distributions along the transport path and beam traces through the focusing system.

Figures 16 and 17 illustrate a typical starting point in the optimization process for one case under investigation. Figure 16 shows the output phase distribution at the end of the last magnetic field for the starting values of the parameters, as squares; the desired end condition is that the distribution lies within the ellipse. Figure 17 shows the beam traces for these starting values. Figures 18 and 19 show the same displays after the algorithm has achieved convergence.

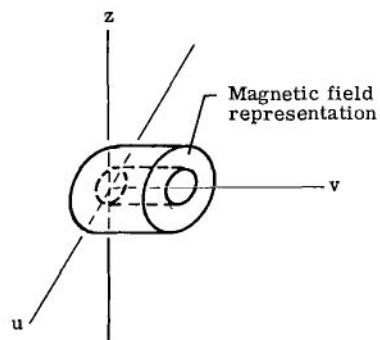
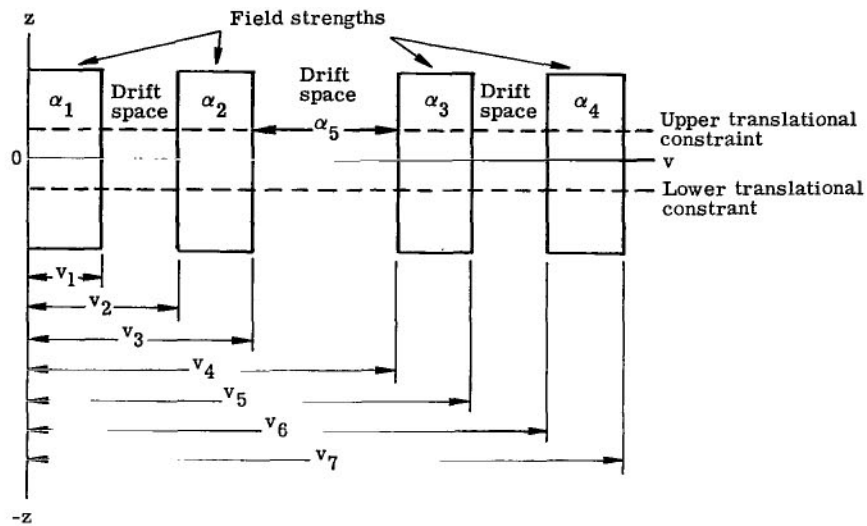


Figure 15.- Geometry of beam-transport problem.

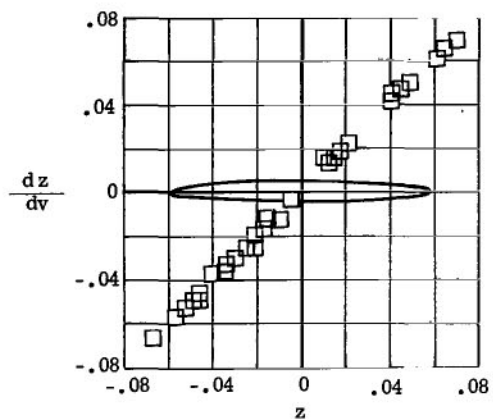
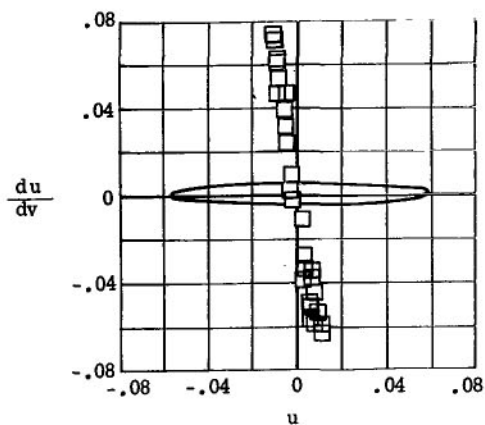
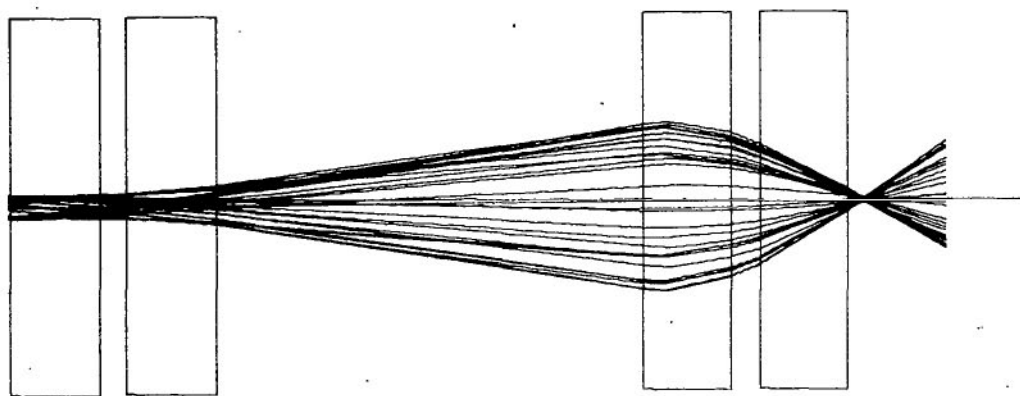
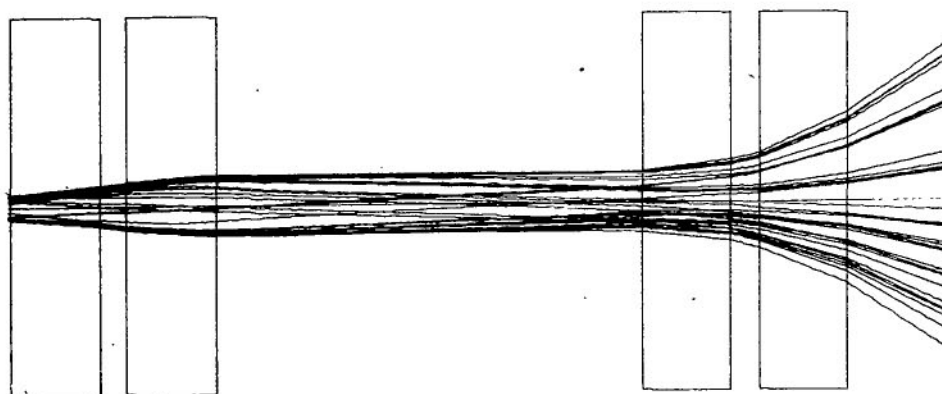


Figure 16.- Phase-plane plots for beam-transport problem. Cost function of 354.7;
 $\alpha_1 = 1.25$; $\alpha_2 = -1.10$; $\alpha_3 = 1.29$; $\alpha_4 = 1.26$; $\alpha_5 = 3.31$.



BOTTOM VIEW



SIDE VIEW

Figure 17.- Beam traces for beam-transport problem. Cost function of 354.7;
 $\alpha_1 = 1.25$; $\alpha_2 = -1.10$; $\alpha_3 = 1.29$; $\alpha_4 = 1.26$; $\alpha_5 = 3.31$.

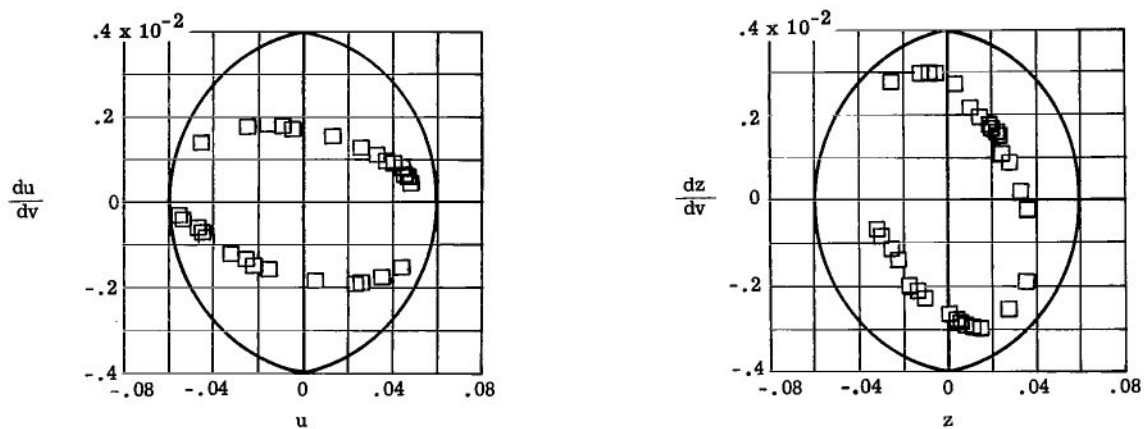


Figure 18.- Phase-plane plots for beam transport problem. Cost function of 0.993;
 $\alpha_1 = 1.06$; $\alpha_2 = -1.18$; $\alpha_3 = 0.89$; $\alpha_4 = -0.77$; $\alpha_5 = 3.82$.

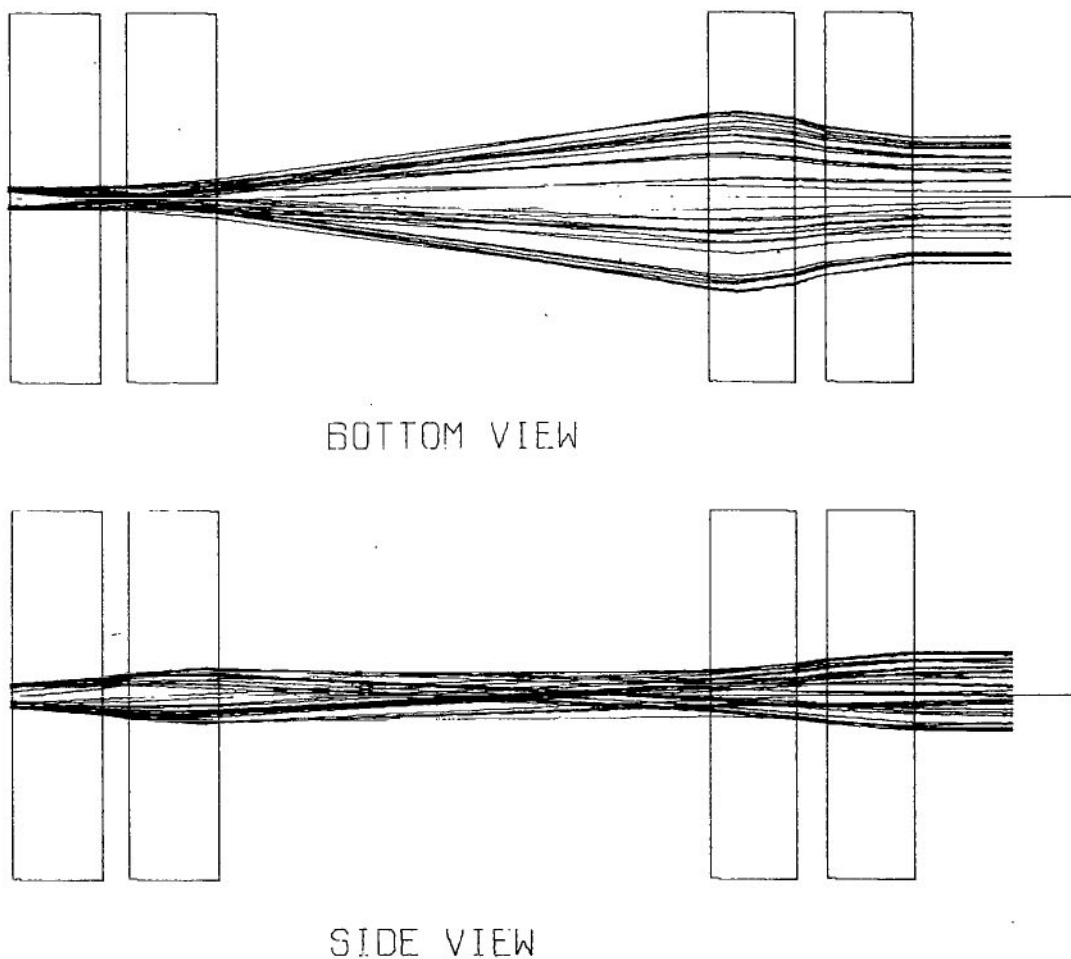


Figure 19.- Beam traces for beam-transport problem. Cost function of 0.993;
 $\alpha_1 = 1.06$; $\alpha_2 = -1.18$; $\alpha_3 = 0.89$; $\alpha_4 = -0.77$; $\alpha_5 = 3.82$.

CONCLUDING REMARKS

A modified version of the creeping-random-search algorithm has been developed and applied to several different types of problems in the field of parameter identification. The modified version, the nonredundant algorithm, reduces the number of cost-function evaluations when compared to the unmodified version, the redundant algorithm. Also, it has been demonstrated that the algorithm can be applied successfully to problems that contain limits, inequality constraints (both dependent and independent, and linear and non-linear), and/or stochastic models; these are problems on which deterministic methods without modification are often of little value.

Other results indicated that strategy performance within the algorithm is highly problem dependent and, therefore, difficult to implement successfully. A method that utilized analyst-algorithm interaction through a cathode-ray-tube display to speed convergence was shown to be feasible.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., November 12, 1971.

APPENDIX A

EQUATIONS FOR HUMAN-TRANSFER-FUNCTION PROBLEM

The equations for the human-transfer-function problem are presented in the following two parts: (1) The transfer-function equations representing the human pilot, presented in Laplace notation, and (2) the equations of motion for two three-degree-of-freedom aircraft in a pursuit environment.

The transfer-function equations are as follows:

$$s\psi_{c,A} = \frac{(\lambda - \psi_A) \frac{a_1}{\alpha_1} \left(1 + \frac{a_2}{\alpha_1} s\right)}{\left(1 + \frac{1}{\alpha_1} s\right)^2}$$

$$s\psi'_{c,A} = \begin{cases} s\psi_{c,A}, & -\alpha_5 \leq s\psi_{c,A} \leq \alpha_5 \\ \alpha_5, & s\psi_{c,A} \geq \alpha_5 \\ -\alpha_5, & s\psi_{c,A} \leq -\alpha_5 \end{cases}$$

$$\phi_{c,A}(s) = \tan^{-1} \left(\frac{s\psi'_{c,A} V_A}{g} \right)$$

$$\delta_a(s) = \frac{(\phi_{c,A} - \phi_A) \frac{\alpha_4}{\alpha_2} \left(1 + \frac{\alpha_3}{\alpha_2} s\right)}{\left(1 + \frac{1}{\alpha_2} s\right)^2}$$

$$\delta'_a(s) = \begin{cases} \delta_a(s), & -c \leq \delta_a \leq c \\ c, & \delta_a \geq c \\ -c, & \delta_a \leq -c \end{cases}$$

$$p_A(s) = \frac{\delta'_a(s)}{1 + K_4 s}$$

$$s\phi_A = p_A(s)$$

APPENDIX A - Concluded

The equations of motion for the aircraft are as follows:

$$\dot{x}_T = V_T \cos \psi_T$$

$$\dot{y}_T = V_T \sin \psi_T$$

$$\dot{x}_A = V_A \cos \psi_A$$

$$\dot{y}_A = V_A \sin \psi_A$$

$$\dot{\psi}_A = \frac{g \tan \phi_A}{V_A}$$

$$\lambda = \tan^{-1} \frac{y_T - y_A}{x_T - x_A}$$

$$R = \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}$$

$$a_z = \frac{1.0}{\cos \phi_A}$$

APPENDIX B

EQUATIONS FOR BEAM-TRANSPORT-DESIGN PROBLEM

The equations for the beam-transport-design problem are presented in the following two parts: (1) The linear differential equations, which are not used in the digital program, and (2) the algebraic solutions to the differential equations, which are used in the program. The equations consist of translational equations which govern motion within a magnetic field and within a drift space, as well as extremum equations which determine extremes of the state variables within the magnetic fields in order to check for violation of the state-variable constraints.

The differential equations within a magnetic field are

$$\frac{d^2u}{dv^2} - \alpha_i u = 0$$

and

$$\frac{d^2z}{dv^2} + \alpha_i z = 0$$

where $i = 1, 2, 3, 4$.

The differential equations within a drift space are

$$\frac{d^2u}{dv^2} = 0$$

and

$$\frac{d^2z}{dv^2} = 0$$

The algebraic equations for magnetic field translation are as follows:

$$u(v_{out}) = u(v_{in}) \cos \left[\sqrt{|\alpha_i|} (v_{out} - v_{in}) \right] + \frac{\frac{du(v_{in})}{dv} \sin \left[\sqrt{|\alpha_i|} (v_{out} - v_{in}) \right]}{\sqrt{|\alpha_i|}}$$

$$\frac{du(v_{out})}{dv} = -u(v_{in}) \sqrt{|\alpha_i|} \sin \left[\sqrt{|\alpha_i|} (v_{out} - v_{in}) \right] + \frac{du(v_{in})}{dv} \cos \left[\sqrt{|\alpha_i|} (v_{out} - v_{in}) \right]$$

$$z(v_{\text{out}}) = z(v_{\text{in}}) \cosh \left[\sqrt{|\alpha_i|} (v_{\text{out}} - v_{\text{in}}) \right] + \frac{\frac{dz(v_{\text{in}})}{dv} \sinh \left[\sqrt{|\alpha_i|} (v_{\text{out}} - v_{\text{in}}) \right]}{\sqrt{|\alpha_i|}}$$

$$\frac{dz(v_{\text{out}})}{dv} = z(v_{\text{in}}) \sqrt{|\alpha_i|} \sinh \left[\sqrt{|\alpha_i|} (v_{\text{out}} - v_{\text{in}}) \right] + \frac{dz(v_{\text{in}})}{dv} \cosh \left[\sqrt{|\alpha_i|} (v_{\text{out}} - v_{\text{in}}) \right]$$

The algebraic equations for drift space translation are as follows:

$$u(v_{\text{out}}) = u(v_{\text{in}}) + \frac{du(v_{\text{in}})}{dv} (v_{\text{out}} - v_{\text{in}})$$

$$\frac{du(v_{\text{out}})}{dv} = \frac{du(v_{\text{in}})}{dv}$$

$$z(v_{\text{out}}) = z(v_{\text{in}}) + \frac{dz(v_{\text{in}})}{dv} (v_{\text{out}} - v_{\text{in}})$$

$$\frac{dz(v_{\text{out}})}{dv} = \frac{dz(v_{\text{in}})}{dv}$$

The algebraic state-variable extremum equations within a magnetic field are as follows:

$$v = \begin{cases} \frac{1}{\sqrt{|\alpha_i|}} \tan^{-1} \left[\frac{\frac{du(v_{\text{in}})}{dv}}{\sqrt{|\alpha_i|} u(v_{\text{in}})} \right], & 0 < v < (v_{\text{out}} - v_{\text{in}}) \\ (v_{\text{out}} - v_{\text{in}}), & v \geq (v_{\text{out}} - v_{\text{in}}) \\ 0, & v \leq 0 \end{cases}$$

$$u(v) = u(v_{\text{in}}) \cos \left[\sqrt{|\alpha_i|} v \right] + \frac{\frac{du(v_{\text{in}})}{dv} \sin \left[\sqrt{|\alpha_i|} v \right]}{\sqrt{|\alpha_i|}}$$

APPENDIX B - Concluded

$$v = \begin{cases} \frac{1}{\sqrt{|\alpha_i|}} \tanh^{-1} \left[\frac{\frac{dz(v_{in})}{dv}}{\sqrt{|\alpha_i|} z(v_{in})} \right], & 0 < v < (v_{out} - v_{in}) \\ (v_{out} - v_{in}), & v \geq (v_{out} - v_{in}) \\ 0, & v \leq 0 \end{cases}$$

$$z(v) = z(v_{in}) \cosh \left[\sqrt{|\alpha_i|} v \right] + \frac{\frac{dz(v_{in})}{dv} \sinh \left[\sqrt{|\alpha_i|} v \right]}{\sqrt{|\alpha_i|}}$$

The algebraic equations are for positive focusing strength α_i . For negative α_i , the equations for u and z are simply interchanged; the u equations become hyperbolic functionals, and the z equations become trigonometric functionals.

REFERENCES

1. Waren, Allan D.; Lasdon, Leon S.; and Suchman, Daniel F.: Optimization in Engineering Design. Proc. IEEE, vol. 55, no. 11, Nov. 1967, pp. 1885-1897.
2. Bekey, George A.: Optimization of Multiparameter Systems by Hybrid Computer Techniques.
Pt. I. Simulation, vol. 2, no. 2, Feb. 1964, pp. 19-32.
Pt. II. Simulation, vol. 2, no. 3, Mar. 1964, pp. 21-29.
3. Gilbert, Elmer G.: A Selected Bibliography on Parameter Optimization Methods Suitable for Hybrid Computation. Simulation, vol. 8, no. 6, June 1967, pp. 350-352.
4. Mitchell, Baker A., Jr.: A Hybrid Analog-Digital Parameter Optimizer for ASTRAC II. AFIPS Conference Proceedings, Vol. 25 - 1964 Spring Joint Computer Conference, Spartan Books, Inc., 1964, pp. 271-285.
5. Bard, Yonathan: Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems. SIAM J. Numerical Anal., vol. 7, no. 1, Mar. 1970, pp. 157-186.
6. Young, Peter C.: Process Parameter Estimation and Self Adaptive Control. Theory of Self-Adaptive Control Systems, P. H. Hammond, ed., Plenum Press, 1966, pp. 118-140.
7. Hague, D. S.; and Glatt, C. R.: A Guide to the Automated Engineering and Scientific Optimization Program - AESOP. Contract NAS 2-4507, Space Div., Boeing Co., [1968]. (Available as NASA CR-73201.)
8. Lasdon, L. S.; Mitter, S. K.; and Waren, A. D.: The Conjugate Gradient Method for Optimal Control Problems. IEEE Trans. Automat. Contr., vol. AC-12, no. 2, Apr. 1967, pp. 132-138.
9. Bryson, Arthur E., Jr.; and Ho, Yu-Chi: Applied Optimal Control. Blaisdell Pub. Co., c.1969.
10. Hadley, G.: Linear Programming. Addison-Wesley Pub. Co., Inc., c.1962.
11. Korn, Granino A.: Hybrid Computer Monte Carlo Techniques. Simulation, vol. 5, no. 4, Oct. 1965, pp. 234-245.
12. White, Ellis: Eastern Simulation Council Meeting. Simulation, vol. 12, no. 2, Feb. 1969, pp. 53-56.
13. Canavos, George C.: A Comparative Analysis of Two Concepts in the Generation of Uniform Pseudo-Random Numbers. Proceedings of 22nd National Conference, A.C.M. Publ. P-67, Thompson Book Co., 1967, pp. 485-501.

14. Feller, William: An Introduction to Probability Theory and Its Applications. Vol. I. John Wiley & Sons, Inc., c.1950.
15. Hadley, G.: Nonlinear and Dynamic Programming. Addison-Wesley Pub. Co., Inc., c.1964.
16. Adams, James J.; and Goode, Maxwell W.: Application of Human Transfer Functions to System Analysis. NASA TN D-5478, 1969.
17. Mehra, Raman K.: Maximum Likelihood Identification of Aircraft Parameters. Eleventh Joint Automatic Control Conference of the American Automatic Control Council, Amer. Soc. Mech. Eng., c.1970, pp. 442-444.
18. Kashyap, R. L.: Maximum Likelihood Identification of Stochastic Linear Systems. IEEE Trans. Automat. Contr., vol. AC-15, no. 1, Feb. 1970, pp. 25-34.
19. Powell, J. David; and Tyler, James S., Jr.: Application of the Kalman Filter and Smoothing to VTOL Parameter and State Estimation. Eleventh Joint Automatic Control Conference of the American Automatic Control Council, Amer. Soc. Mech. Eng., c.1970, pp. 449-450.
20. Steffen, Klaus G.: High Energy Beam Optics. Interscience Publ., c.1965.
21. Bowker, Albert H.; and Lieberman, Gerald J.: Engineering Statistics. Prentice-Hall, Inc., c.1959.
22. Parrish, Russell V.; Dieudonne, James E.; and Filippas, Tassos A.: Application of Monte Carlo Techniques to Optimization of High-Energy Beam Transport in a Stochastic Environment. NASA TN D-6571, 1971.

TABLE I.- COMPARISON OF NUMBER OF TRIES FOR BOTH
ALGORITHMS BEFORE STEP-SIZE CUT

Number of parameters, n	Nonredundant tries, r (a)	Redundant tries, r (P > 90 percent) (b)	<u>Redundant tries</u> Nonredundant tries
1	2	9	4.50
2	8	39	4.88
3	26	149	5.73
4	80	540	6.75
5	242	1 891	7.81
6	728	6 480	8.90
7	2186	21 850	10.00

$$^a r = 3^n - 1.$$

$$^b P = \sum_{k=0}^{3^n} (-1)^k \binom{3^n}{k} \left(1 - \frac{k}{3^n}\right)^r.$$

TABLE II.- STRATEGY DEFINITIONS

Pure random	Strategy 1
Last successful direction first	Strategy 2
Variable step size based on runs	Strategy 3
Correlation on past history	Strategy 4
Correlation on last success	Strategy 5
Strategies 2 and 4 combined	Strategy 6
Strategies 2 and 5 combined	Strategy 7
Strategies 2 and 3 combined	Strategy 8
Strategies 3 and 4 combined	Strategy 9
Strategies 3 and 5 combined	Strategy 10
Strategies 2, 3, and 4 combined	Strategy 11
Strategies 2, 3, and 5 combined	Strategy 12

TABLE III.- STRATEGY RANKINGS

Strategy	Static problem I		Static problem II		Dynamic problem I	
	Rank	Average number of iterations to convergence	Rank	Average number of iterations to convergence	Rank	Average number of iterations to convergence
8	2	44	2	58	1	33
12	1	35	1	45	6	59
11	4	56	3	60	3	48
10	3	51	4	64	5	54
9	10	175	5	70	2	38
3	5	66	10	269	3	48
7	8	143	7	153	7	156
6	6	97	8	168	8	161
2	9	156	6	132	9	169
5	7	118	9	200	11	247
1	11	198	12	307	10	233
4	12	298	11	280	12	423

TABLE IV.- RUN PARAMETERS

[S = Number of successes in a row before step increase;
F = Number of failures in a row before step decrease]

F	Average number of iterations to convergence when -					
	S = 1	S = 2	S = 3	S = 4	S = 5	S = 6
Static problem I						
1	113.6	88.7	68.7	68.1	49.7	63.8
2	80.9	83.2	138	128.8	175.6	181.7
3	84.3	71.8	79.9	75.3	87.6	80.7
4	104.3	77.7	80.0	82.2	79.9	78.8
5	122.3	88.2	94.9	89.0	87.2	86.3
6	112.7	99.1	100.9	96.8	93.4	93.9
Static problem II						
1	772.6	326.5	184.5	146.0	145.7	130.3
2	166.0	1694.6	1878.6	803.8	763.0	659.9
3	93.3	265.9	1438.1	4930.4	4700.1	2720.2
4	99.1	258.6	264.7	829.9	1112.2	1853.9
5	118.7	232.2	235.5	302.2	398.1	795.3
6	121.5	255.3	249.8	269.9	248.4	410.0

TABLE V.- COMPARISON OF ALGORITHMS FOR
HUMAN-TRANSFER-FUNCTION PROBLEMS

Parameters	True value	Creeping search		Steepest descent		Davidon	
		Starting value	Final value	Starting value	Final value	Starting value	Final value
α_1	14.75	18.00	14.75	18.00	15.483	18.00	16.443
α_2	10.0	13.00	10.0	13.00	12.235	13.00	13.147
α_3	26.5	30.00	26.5	30.00	29.112	30.00	29.950
α_4	25.4	20.00	25.4	20.00	27.497	20.00	19.995
α_5	.1	.17	.1	.17	.09678	.17	.09847
Cost	----	0.65814	0	0.65814	0.001736	0.65814	0.00603
No. cost-function evaluations . . .			2723		3652		4384
Total computer time			1743		1828		2210



017 001 C1 U 08 711217 S00903DS
DEPT OF THE AIR FORCE
AF WEAPONS LAB (AFSC)
TECH LIBRARY/WLOL/
ATTN: E LOU BOWMAN, CHIEF
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546**